An Approach to Syndrome Differentiation in Traditional Chinese Medicine based on Neural Network

Minghui Shi
Institute of Artificial Intelligence, Xiamen University, Xiamen, Fujian, 361005, China
smh@xmu.edu.cn

Changle Zhou
Institute of Artificial Intelligence, Xiamen University, Xiamen, Fujian, 361005, China
dozero@xmu.edu.cn

Abstract

Although the traditional knowledge representation based on rules is simple and explicit, it is not effective in the field of syndrome differentiation in Traditional Chinese Medicine (TCM), which involves many uncertain concepts. To represent uncertain knowledge of syndrome differentiation in TCM, two methods were presented respectively based on certainty factors and certainty intervals. Exploiting these two methods, an approach to syndrome differentiation in TCM was proposed based on neural networks to avoid some limitations of other approaches. The main advantage of the approach is that it may realize uncertain inference of syndrome differentiation in TCM, whereas it doesn't request experts to provide all possible combinations for certainty degrees of symptoms and syndromes. Rather than Back Propagation (BP) algorithm but its modification was employed to improve the capability of generalization of neural networks. First, the standard feedforward multilayer BP neural network and its modification were introduced. Next, two methods for knowledge representation, respectively based on certainty factors and certainty intervals, were presented. Then, the algorithm was proposed based on neural network for the uncertain inference of syndrome differentiation in TCM. Finally, an example was demonstrated to illustrate the algorithm.

1. Introduction

Intelligent diagnosis for Traditional Chinese Medicine (TCM) is becoming a promising research area. In 1980s, some expert systems for TCM based on rules were developed in China. Those expert systems, however, cannot show perfect performance due to their inherit limitations for dealing with vague concepts and the complexity of diagnosis in TCM. Since early 1990s, artificial neural networks (ANNs) have been applied to this area, and show a valuable research direction.

In the theory of TCM, *symptoms* refer to various abnormal manifestations of a disease, while *syndrome* is a summarization of the development of a disease at a certain stage, including cause, location, nature, pathogenesis and the relevant symptoms and signs. Take "*kidney yin asthenia*" for example. It suggests that the location is in the kidney; the nature is *yin asthenia*. Diagnosis in TCM is mainly composed of two aspects, namely, *diagnostic methods* and *syndrome differentiation* [1]. *Diagnostic methods* are used to examine patients and collect *symptoms*, mainly including inspection, listening and olfaction, inquiry and pulse-taking, known as the *four diagnostic methods*. *Syndrome differentiation* means to synthesize and analyze the pathological data so as to decide the nature of *syndrome*, and is the method for understanding the nature of disease. *Diagnostic methods* serve for *syndrome differentiation*, and *syndrome differentiation* is the crucial part of the entire diagnostic process.

Since *symptoms* are evidences for *syndrome differentiation*, from the viewpoint of artificial intelligence (AI), *syndrome differentiation* systems may be viewed as tools mapping *symptoms* into *syndromes*. ANNs also can be viewed as mapping tools [2]. Recently, there have been some published researches on *syndrome differentiation* in TCM using ANNs [3-5] in China. Most of them assume that the space of symptoms is a two-value space, and use variants X_i (i=1,2,…,n) denote symptoms, so a group of symptoms can be denoted by a set: $\{ X_i \mid i=1,2,\ldots,n \}$, and the set of symptoms of each patient corresponds to one vector $(X_1, X_2, \ldots, X_n)$, where the value of variant X_i is 1 when the i-th symptom occurs, otherwise is 0, and can be viewed as a point in n-dimensional space.

Unfortunately, *syndrome differentiation* in TCM is a complicated process. The two-value method cannot exactly represent the knowledge about *syndrome*

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

differentiation in TCM, since patients' symptoms and doctors' diagnosis are matters with strong vagueness and uncertainty. For this reason, some people attempt to apply fuzzy mathematics to the inference of *syndrome differentiation* in TCM [6]. This method, however, meets difficulty for the determination of fuzzy sets, fuzzy matrix and measurement of similarity between fuzzy sets, etc.

In this paper, we propose a simple, but effective and powerful approach to *syndrome differentiation* in TCM by combining the method based on production rules with that based on ANNs. The fundamental idea is: to represent uncertain knowledge of *syndrome differentiation* in TCM, we attach typical *certainty factors* or *certainty intervals* to the concepts in the production rules; to avoid the inefficient matching inference of production rules, we exploit ANNs rather than production rules to implement inference process, since ANNs can realize the function of the production rules attached *certainty factors* or *certainty intervals*.

This paper is organized as follows. In the next section, we introduce the back propagation (BP) neural network and its modification. In section 3, to represent uncertain knowledge of *syndrome differentiation*, the two methods, based on *certainty factors* and *certainty intervals* respectively, are presented. The algorithm for the uncertainty inference of *syndrome differentiation* is proposed based on neural networks in section 4. To illustrate the algorithm, an example is given in section 5. We conclude this paper in section 6.

2. BP Neural Network and its Modification

In general, the structure of BP neural network [7] is a three-layer feedforward neural network. Honik et al. [8] established that a feedforward multilayer neural network that has only one hidden layer, with a sufficient number of neurons, acts as a universal approximation of nonlinear mappings. However, an "over-designed" architecture will tend to "overfit" the training data [9], which results in the loss of the generalization property of the neural network.

Here we introduce an effective method called *regularization*, which can significantly improve the generalization property of neural work by reducing the number of active weights and biases. By means of *regularization*, we can improve the generalization property of neural network, while need not determine the scale of neural network accurately. The main difference between *regularization* and the standard BP method is that the cost function adopted is different. The standard BP neural network uses *mse* in (1),

$$mse = \frac{1}{n_o}\sum_{i=1}^{n_o}(e_i)^2 = \frac{1}{n_o}\sum_{i=1}^{n_o}(t_i - y_i)^2 , \qquad (1)$$

whereas *regularization* uses the regularized cost function, which can be written as follows:

$$msereg = \gamma \times mse + (1-\gamma)\times msw , \qquad (2)$$

where γ is a performance parameter, and

$$msw = \frac{1}{n}\sum_{j=1}^{n} w_j^2 , \qquad (3)$$

where n is the number of all the weights and biases, *msw* is the mean square of all the weights and biases. Usually it is difficult to determine the value of the performance parameter γ. If it were too large, the generalization property would become bad; on the contrary, if it were too small, mapping property would become bad. Fortunately, a function named '*trainbr*' is provided by the neural network toolbox in MatLab7.0, which can set automatically the optimization value of the performance parameter γ.

3. Representation of Uncertain Knowledge for Syndrome Differentiation

In this section, we present two methods of knowledge representation, respectively based on *certainty factors* and *certainty intervals*.

Although the traditional knowledge representation based on rules is simple and explicit, it is not effective in the field of *syndrome differentiation* in TCM, which involves many uncertain concepts. For example, there is a diagnostic law in the theory of tongue inspection in TCM: *A man whose tongue is black and dry must suffer the syndrome of kidney yin asthenia*. One rule can be extracted from the experience as follows.

Rule_1: IF *tongue is black* AND *tongue is dry* THEN *kidney yin asthenia*.

Rule_1 is not applicable for intelligent simulation based on rules, since "*black*", "*dry*" and "*yin asthenia*" are vague and uncertain concepts.

To make the rule-based knowledge more tractable, we attach every uncertain concept with a certainty factor (CF), which means the degree of the concept's certainty, e.g., Rule_1 with CFs is shown as follows.

Rule_2: IF *tongue is black* (CF_1) AND *tongue is dry* (CF_2) THEN *kidney yin asthenia* (CF_3),

where CF_i (i=1, 2, 3) are certain factors, values of which refer to the certain degrees of the corresponding symptoms or syndrome, defined by the formula (4).

$$CF_i = \begin{cases} 0 & \text{absolute uncertainty} \\ 1 & \text{absolute certainty} \\ \in(0,1) & \text{others} \end{cases} \quad (4)$$

Rule_2 means: if the certainty degree of *blackish tongue* is CF_1, and that of *dry tongue* is CF_2, then the certainty degree of *kidney yin asthenia* is CF_3.

In TCM, however, there exists another type of uncertain concept such as "*70 percent to 80 percent*" or "*eight to nine out of ten*". Similarly, we can express such information by means of *certainty intervals*, and one certainty interval can be represented by two *certainty factors*, which respectively means the lower limit or the upper limit of the certainty interval.

For example, rule_1 may be rewritten with *certainty intervals* as follows:

Rule_3: IF *tongue is black* (CF_1^L, CF_1^U) AND *tongue is dry* (CF_2^L, CF_2^U) THEN *kidney yin asthenia* (CF_3^L, CF_3^U),

where superscript *L* denotes lower limit, superscript *U* denotes upper limit. For example, (CF_1^L, CF_1^U) denotes the certainty interval [CF_1^L, CF_1^U].

Without loss of generality, for the method based on *certainty factors*, we may express knowledge for syndrome differentiation by the following form:

Rule_4: IF X_1 (CF_{X_1}) AND X_2 (CF_{X_2}) AND … AND X_n (CF_{X_n}) THEN Y_j (CF_{Y_j}),

where X_i (i=1,2,…,n) denote symptoms; Y_j (j=1,2…m) denote syndromes, m is the number of possible syndromes. Similarly, for the method based on *certainty intervals*, we may the following form:

Rule_5: IF X_1 ($CF_{X_1}^L$, $CF_{X_1}^U$) AND X_2 ($CF_{X_2}^L$, $CF_{X_2}^U$) AND … AND X_n ($CF_{X_n}^L$, $CF_{X_n}^U$) THEN Y_j ($CF_{Y_j}^L$, $CF_{Y_j}^U$).

Attached by *certainty factors* or *certainty intervals*, the vague and uncertain concepts are quantified and clarified. So the rules based on certainty degrees (*certainty factors* or *certainty intervals*) become more tractable. However, some problems arise: 1) we couldn't enumerate all the values of *certainty factors* or *certainty intervals*; 2) since one rule is replaced by several rules with certainty degrees, the number of rules increases rapidly; 3) knowledge management would become difficult and inference process would become very slow.

To solve these problems, in the next section, we propose an approach which can realize the function of rules while avoiding their shortcomings.

4. Algorithm Based on Neural Network

In this section, the algorithm for the uncertainty inference of *syndrome differentiation* in TCM is proposed based on the modified BP neural network. To illustrate it, a demonstration will be shown in section 5.

The details for the algorithm are outlined as follows.

Step 1: knowledge acquisition and representation.

Rules in the form of rule_1 are used to represent knowledge, which may be obtained by the analysis of materials or from experts' experiences. These rules constitute the initial knowledge base.

Step 2: formalization.

The symptoms and syndromes in the knowledge base should be formalized with different variants. Every variant denotes a symptom or a syndrome. For example, variant X_i (i=1, 2, … , n) denote symptoms; Y_j (j=1, 2, … , m) denote syndromes .

Step 3: obtain typical combinations of certain degree.

We should get several typical combinations of certain degrees (*certainty factors* or *certainty intervals*) of symptoms and syndromes with the help of experts or by studying clinical cases. The initial rules can be rewritten in form of rule_4 or rule_5. Furthermore we assign variants with corresponding values as follows:

i) For the combination of *certainty factors*,

$$X_i = CF_{X_i}, \; Y_j = CF_{Y_j},$$

where CF_{X_i} is the certainty factor of symptom X_i; CF_{Y_j} is the certainty factor of syndrome Y_j.

ii) For the combination of *certainty intervals*,

$$X_i = [CF_{X_i}^L, \; CF_{X_i}^U], \quad Y_j = [CF_{Y_j}^L, \; CF_{Y_j}^U],$$

where $[CF_{X_i}^L, CF_{X_i}^U]$ means the certainty factor of symptom X_i must belong to the interval [$CF_{X_i}^L$, $CF_{X_i}^U$]; [$CF_{Y_j}^L$, $CF_{Y_j}^U$] means the certainty factor of Y_j must belong to the interval [$CF_{Y_j}^L$, $CF_{Y_j}^U$].

Step4: produce training samples and test samples.

We should obtain some samples for training the neural networks and other samples for testing the trained neural networks. More detail interpretations will be shown in section 5.

Step 5: design the neural network.

Step 6: train the neural network.

By using the training samples produced in step 4, we train the neural network so as to make the neural network learn the rules in the knowledge base.

Step7: testing the neural network.

By means of the test samples produced in step 4, we test the neural network for its effectiveness of learning and capability of generalization.

To design the structure of the neural networks in step 5, can adopt the method based on either *certainty factors* (Fig. 1 (a) and (c)) or *certainty intervals* (Fig. 1 (b) and (d)). Furthermore, according to the relevant degree between syndromes, these two methods can be respectively used by two ways: first way: diagnosis for one syndrome (Fig. 1 (a) and (b)), and second way: diagnosis for several syndromes (Fig. 1 (c) and (d)). The main difference between the first way and the second way is the number of syndromes in the output layer. There are several syndromes for the latter, while only one for the former.

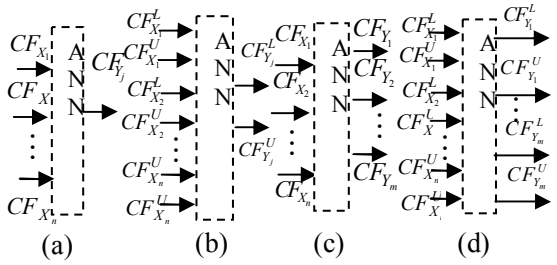

Fig.1. Model based on certainty factor or certainty interval

5. Example

To illustrate the algorithm, we take an example by exploiting the neural network toolbox in Matlab 7.0. For simplicity, we only present a simplified example for the method based on *certainty factors* and taking the first way such as Fig. 1 (a), since other situations are just extensions for it.

Step 1: knowledge acquisition and representation.

Without loss of generalization, we employ rule_1 as initial knowledge.

Step 2: formalization.

The symptoms ("*tongue is black*" and "*tongue is dry*") and the syndrome ("*kidney yin asthenia*") in the rule_1 are formalized as shown in Table 1.

Table 1. Formaliztion of symptoms and syndrome in rule_1

symptom		syndrome
tongue is black	*tongue is dry*	*tongue is black*
X_1	X_2	Y_1

Step 3: obtain typical combinations of certain degree.

Without loss of generalization, we assume that the combinations of the *certainty factors* between the symptoms and the syndrome in rule_1 have been obtained from experts as shown in Table 2. We use the data in Table 2 so as to easily find the rule hidden in them about the relationship between *certainty factors* of symptoms and syndromes, so that we can examine the algorithm's effectiveness, i.e., we can examine whether the neural network really can learn the hidden law and apply the law to other situations.

Table 2. Combinations of the certainty factors

number	symptom		syndrome
	tongue is balck	*tongue is dry*	*kidney yin asthenia*
	X_1	X_2	Y_1
	CF_{X_1}	CF_{X_2}	CF_{Y_j}
1	0.2	0.2	0.2
2	0.4	0.4	0.4
3	0.5	0.5	0.5
4	0.7	0.7	0.7
5	0.8	0.8	0.8

Step4: produce training samples and test samples.

We use number 1, 3 and 5 as training samples, while number 2 and 4 as test samples. Thus we produce the following input and target for training and testing.

Training sample:

$train\_input = [0.2\ 0.2;\ 0.5\ 0.5;\ 0.8\ 0.8]^T$

$train\_target = [0.2\ 0.5\ 0.8]$

Testing sample:

$test\_input = [0.4\ 0.4;\ 0.7\ 0.7]^T$

$test\_target = [0.4\ 0.7]$

Step 5: design the neural network.

We employ a three-layer neural network shown in Fig. 2. The number of neurons in the input layer is 2, the number of neurons in the output layer is 1. Since there is no restrict limitation to determine the number of neurons in hidden layer, we take 5 hidden neurons. We use the function *transig* as the transfer function in the hidden layer, and *logsig* in the output layer. The reason that we use *logsig* is because its output belongs to the interval [0, 1], which is just the value range of *certainty factors*.

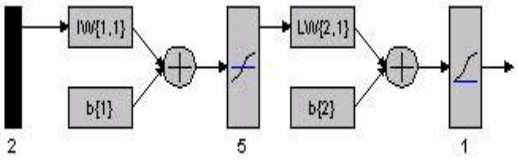

Fig. 2. Structure of designed neural network

Step 6: train the neural network.

We train the neural network using the training sample produced in step 4. The values of the training parameters are adopted with the default values in MATLAB7.0 except for the training function '*trainbr*' , which is provided by the neural network toolbox in MatLab7.0, to improve the capability of generalization by *regularization* presented in section 2.

The main sentence used in the program in MatLab 7.0 is as follows:

0-7695-2875-9/07 $25.00

net = newff ([0 1;0 1], [5 1], {'*tansig*' '*purelin*'}, '*trainbr*') ;

Step7: test the neural network

The trained neural network may be tested by two phases. In the first phase, the *train_input* of the training sample is used as the input of the neural network. The actual output of the neural network is [0.1891 0.4799 0.7533], which is close to the *train_target*. In the second phase, the *test_input* of the testing sample is used as the input of the neural network. The actual output of the neural network is [0.3844 0.6648], which is close to the *test_target*. The results show that the neural network has not only learned the law hidden in the data numbered 1, 3 and 5 in Table 2, which represent the typical combinations of *certainty factors* between symptoms and syndrome, but also has the capability of generalization to apply the learned law to other situations with new data numbered 2 and 4 in Table 2.

6. Conclusion

Production rules with *certainty factors* or *certainty intervals* can represent uncertain knowledge for *syndrome differentiation* in TCM, while they have some limitations such as difficult knowledge management and inefficient inference, since the values of *certainty factors* or *certainty intervals* may be innumerable. To solve these problems, in this paper, an approach to *syndrome differentiation* in TCM is proposed based on a modified BP neural network. The results of simulation show that neural networks can not only learn experts' experiences, but also have the capability for applying the learned knowledge to more general situations. Thus it is not necessary for experts to give all kinds of combinations of certainty values of symptoms and syndromes. What they should do is only to give some typical combinations of certainty degrees. Two methods are presented respectively based on *certainty factors* or *certainty intervals*. In fact, they can be combined and become hybrid methods. That is, we may use *certainty factors* for symptoms, while *certainty intervals* for syndromes, and vice versa. Whatever methods we would adopt, the key is to obtain representative samples reflecting the hidden relationship between symptoms and syndromes.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. 60672018), the National High Technology Research and Development Program of China (863 Program) (Grant No. 2006AA01Z129), and the Innovation Foundation for Science and Technology of Xiamen University of China (Grant No. XDKJCX20063011).

References

[1] L. F. Wang, Z. G. Li and B. Bao, *Diagnosis of Traditional Chinese Medicine*, Publication House of Shanghai University of Traditional Chinese Medicine (in English-Chinese), 2002, pp. 1-2.

[2] F. M. Ham and I. Kostanic, *Principles of Neurocomputing for Science & Engineering*, McGraw-Hill Companies, Inc., 2001, pp. 96-97.

[3] L. Chen and W. M. Li, "Expert System Based on Neural Networks for TCM Diagnosis and Treatment of Hypertension", Vol. 20(2), Computer Engineering (in Chinese), 1994, pp. 23-26.

[4] Z. J. Zhou, Z. Y. Mao, and Z. Z. Deng, "Neural Network Approach to Medical Grading of Rheumatoid Arthritis", Vol. 16(4), Journal of Biomedical Engineering (in Chinese), 1999, pp. 379-482.

[5] F. W. Xu and K. B. Cai, "Application of Neural Networks to the Detection of Pulse Signals", Vol. 27(8), Journal of Chongqing University (in Chinese), 2004, pp. 35-39.

[6] L. Li, L. Xu, D. H. Li, etc., "Study on the Application of Fuzzy Mathematics Method in Traditional Chinese Medicine Research", Vol. 11(6), Clinical Medical Journal of China. 2004, pp. 934-936.

[7] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning Representations by Back-Propagating Errors", Vol. 323, Nature , 1986, pp. 533-536.

[8] K. Hornik, M. Stinchcombe and H. White, "Multilayer Feedforward Networks Are Universal Approximators", Vol. 2(5), Neural Networks, 1989, pp. 359-366.

[9] F. M. Ham, "Detection and Classification of Biological Substances Using Infrared Absorption Spectroscopy and Hybrid Artificial Network", Vol. 1(1), Journal of Artificial Networks, 1994, pp. 101-104.